Analysis of Unicorn Startups

Contents

1 Setup

1.1 Import Packages

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.ticker import FuncFormatter
import seaborn as sns
```

2 Data Preparation

2.1 Load Data

```python
pd.set_option('display.max_columns', 50, 'display.width', 200)
df = pd.read_csv('input/datasets/Unicorns_Completed (2024).csv')
```

2.2 Data Cleaning

```python
import re
def convert_years_months(s):
    m = re.match(r'(\d+)y?\s?(\d+)m?o?', s)
    return f'{m[1]}y{m[2]}m' if m else s

df['Years to Unicorn'] = df['Years to Unicorn'].apply(convert_years_months)

def correct_industry_labels(s):
    if s == 'Health':
        return 'Healthcare & Life Sciences'
    if s == 'West Palm Beach':
        return 'Enterprise Tech'
    return s

df['Industry'] = df['Industry'].apply(correct_industry_labels)
```

2.3 Prepare Data

```python
df['Unicorn Date'] = pd.to_datetime(df['Unicorn Date'])
df['Valuation ($B)'] = pd.to_numeric(df['Valuation ($B)'])
df['Unicorn Year'] = df['Unicorn Date'].dt.year
df['Funding ($B)'] = df['Total Equity Funding ($)'] / 1e9
```

2.4 Preview Data

```
df.head()
```

3 Descriptive Analysis

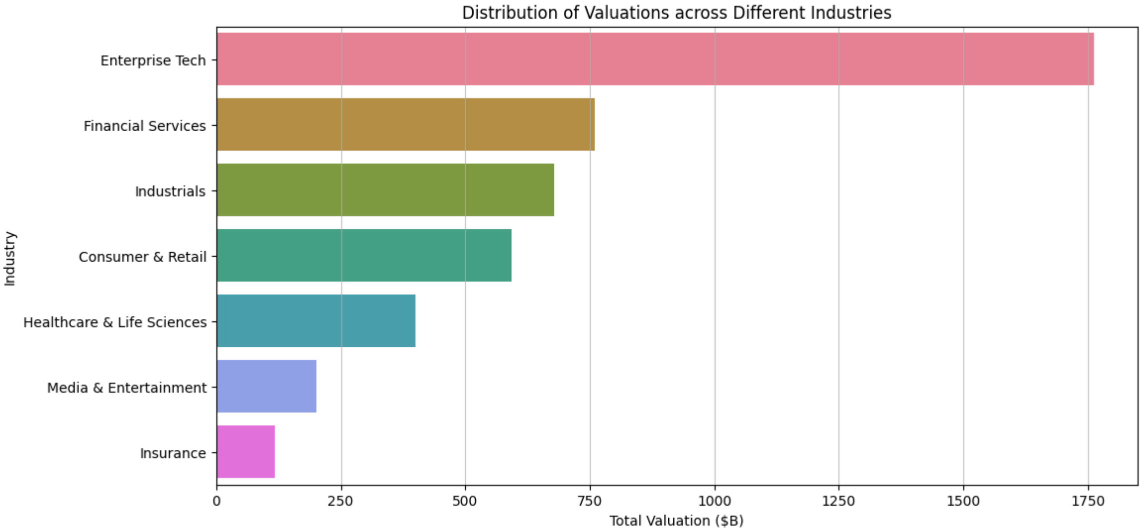
3.1 Distribution

3.1.1 Valuations

Distribution of Valuations across Different Industries

```
# Group by industry and sum valuations
industry_valuation_df = df.groupby('Industry')['Valuation
↪  ($B)'].sum().reset_index().sort_values('Valuation ($B)', ascending=False)
industry_valuation_df
```

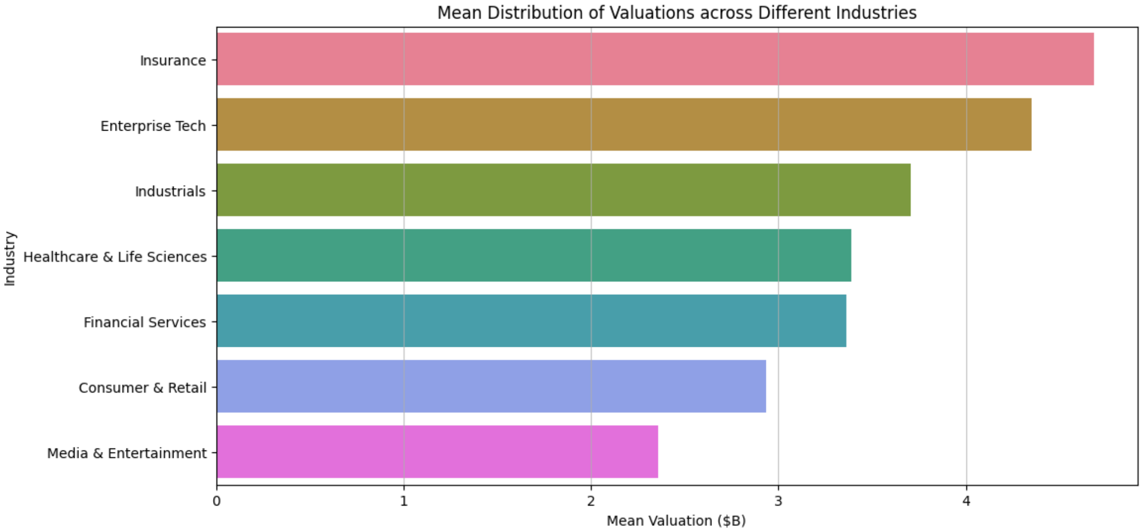
```
plt.figure(figsize=(12, 6))
sns.barplot(y=industry_valuation_df['Industry'], x=industry_valuation_df['Valuation ($B)'],
↪  hue=industry_valuation_df['Industry'], palette='husl')
plt.title('Distribution of Valuations across Different Industries')
plt.xlabel('Total Valuation ($B)')
plt.ylabel('Industry')
plt.grid(axis='x', alpha=0.75)
```



Mean Distribution of Valuations across Different Industries

```
# Group by industry and sum valuations
industry_valuation_df = df.groupby('Industry')['Valuation
↪  ($B)'].mean().reset_index().sort_values('Valuation ($B)', ascending=False)
industry_valuation_df
```

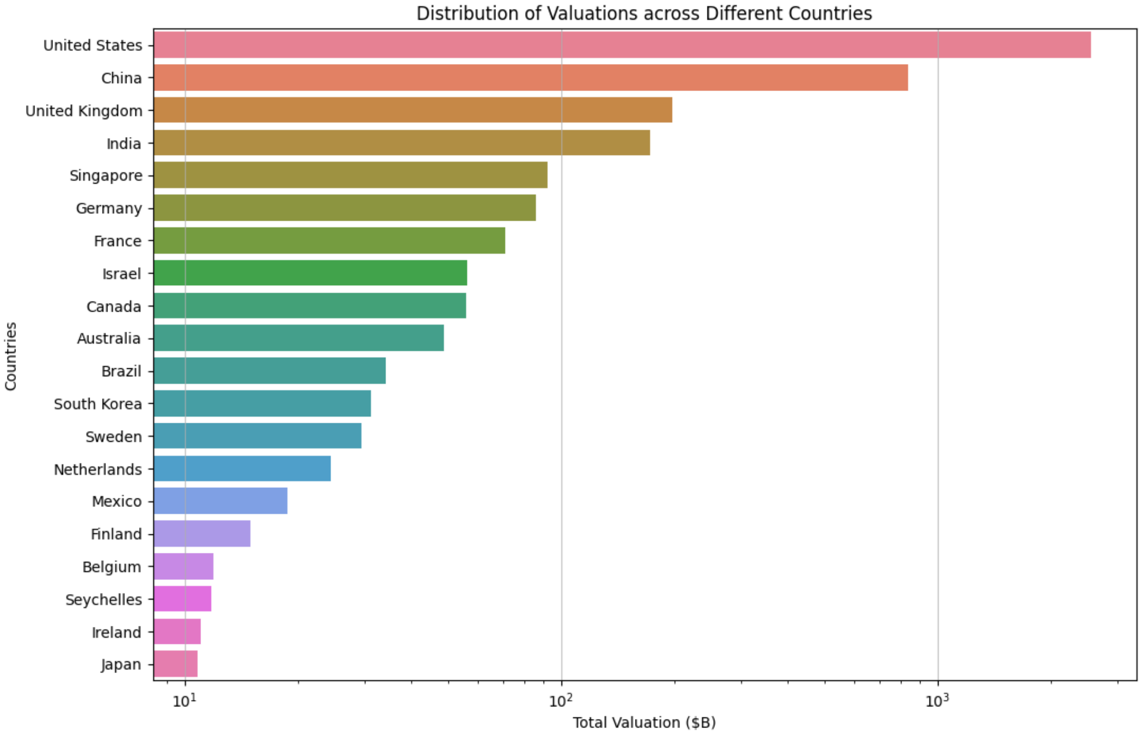
```
plt.figure(figsize=(12, 6))
sns.barplot(y=industry_valuation_df['Industry'], x=industry_valuation_df['Valuation ($B)'],
→  palette='husl', hue=industry_valuation_df['Industry'])
plt.title('Mean Distribution of Valuations across Different Industries')
plt.xlabel('Mean Valuation ($B)')
plt.ylabel('Industry')
plt.grid(axis='x', alpha=0.75)
```



Distribution of Valuations across Different Countries

```
# Group by Country and sum valuations
country_valuation_df = df.groupby('Country')['Valuation
→  ($B)'].sum().reset_index().sort_values('Valuation ($B)', ascending=False).head(20)
country_valuation_df
```

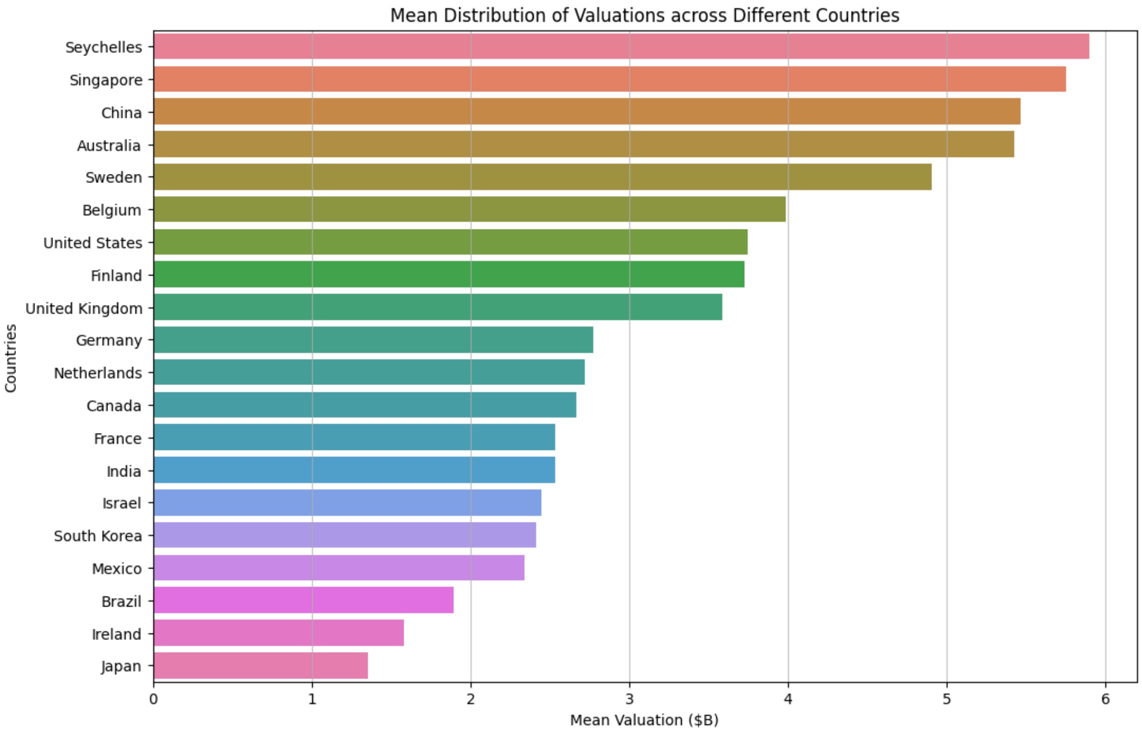
```
plt.figure(figsize=(12, 8))
sns.barplot(y=country_valuation_df['Country'], x=country_valuation_df['Valuation ($B)'],
→  palette='husl', hue=country_valuation_df['Country'])
plt.title('Distribution of Valuations across Different Countries')
plt.xlabel('Total Valuation ($B)')
plt.ylabel('Countries')
plt.grid(axis='x', alpha=0.75)
plt.xscale('log')
plt.show()
```



Mean Distribution of Valuations across Different Countries

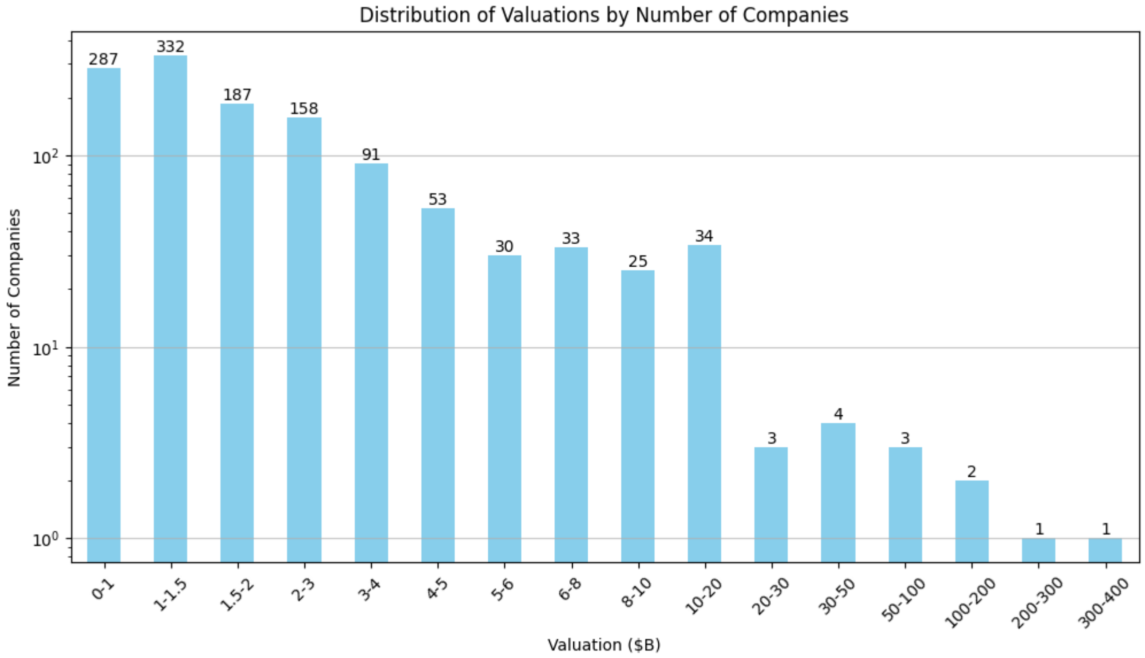
```
mean_country_valuation_df =
↪  df[df['Country'].isin(country_valuation_df['Country'])].groupby('Country')['Valuation
↪  ($B)'].mean().reset_index().sort_values('Valuation ($B)', ascending=False).head(20)
mean_country_valuation_df
```

```
plt.figure(figsize=(12, 8))
sns.barplot(y=mean_country_valuation_df['Country'], x=mean_country_valuation_df['Valuation
↪  ($B)'], palette='husl', hue=mean_country_valuation_df['Country'])
plt.title('Mean Distribution of Valuations across Different Countries')
plt.xlabel('Mean Valuation ($B)')
plt.ylabel('Countries')
plt.grid(axis='x', alpha=0.75)
plt.show()
```



Distribution of Valuations by Number of Companies

```
# Define the bins for valuation ranges
bins = [0, 1, 1.5, 2, 3, 4, 5, 6, 8, 10, 20, 30, 50, 100, 200, 300, 400]
labels =  [f'{a}-{b}' for a, b in zip(bins[:-1], bins[1:])]
cuts = pd.cut(df['Valuation ($B)'], bins=bins, labels=labels)

# Count the number of companies in each bin
valuation_distribution = cuts.value_counts().sort_index()

# Plot the Bar Chart
plt.figure(figsize=(12, 6))
ax = valuation_distribution.plot(kind='bar', color='skyblue')
ax.bar_label(ax.containers[0])
plt.title('Distribution of Valuations by Number of Companies')
plt.xlabel('Valuation ($B)')
plt.ylabel('Number of Companies')
plt.xticks(rotation=45)
plt.grid(axis='y', alpha=0.75)
plt.yscale('log')
plt.show()
```

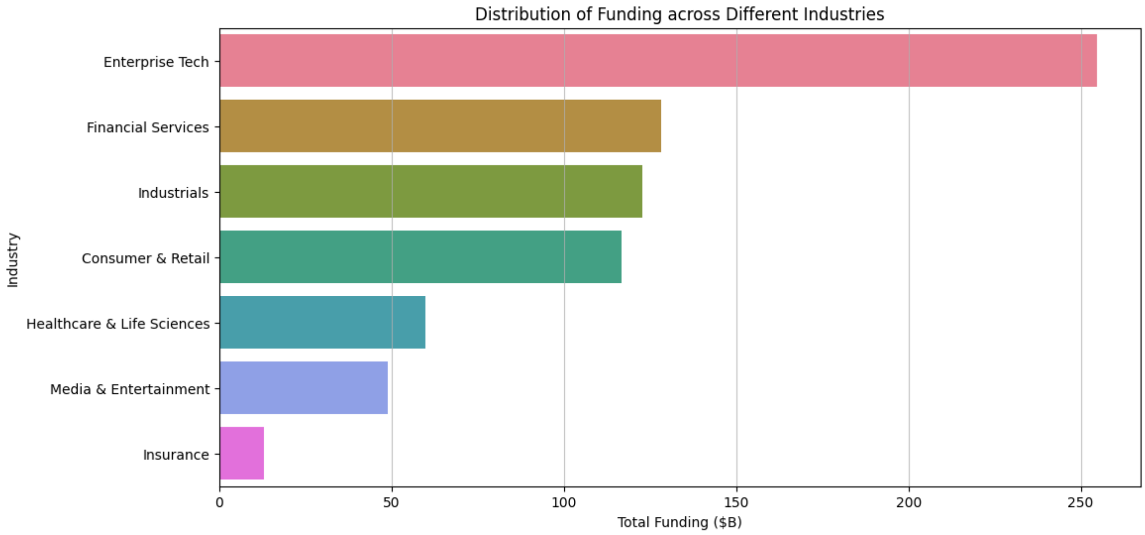


3.1.2 Funding

Distribution of Funding across Different Industries

```
# Group by industry and sum valuations
industry_funding_df = df.groupby('Industry')['Funding
↪  ($B)'].sum().reset_index().sort_values('Funding ($B)', ascending=False)
industry_funding_df
```

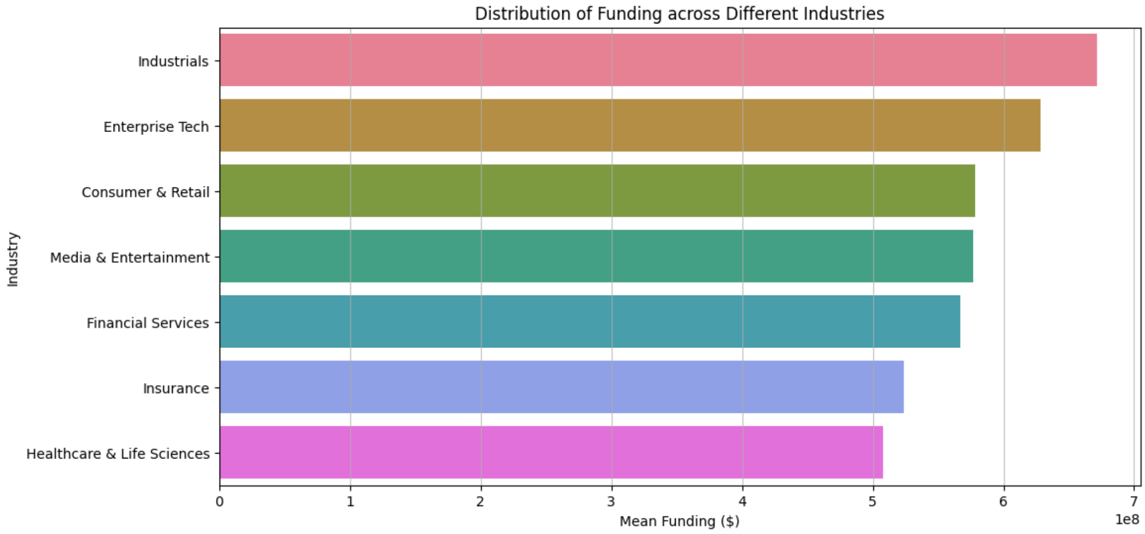
```
plt.figure(figsize=(12, 6))
sns.barplot(y=industry_funding_df['Industry'], x=industry_funding_df['Funding ($B)'],
↪  palette='husl', hue=industry_funding_df['Industry'])
plt.title('Distribution of Funding across Different Industries')
plt.xlabel('Total Funding ($B)')
plt.ylabel('Industry')
plt.grid(axis='x', alpha=0.75)
```



Mean Distribution of Funding across Different Industries

```
industry_funding_df = df.groupby('Industry')['Total Equity Funding
↪ ($)'].mean().reset_index().sort_values('Total Equity Funding ($)', ascending=False)
industry_funding_df
```

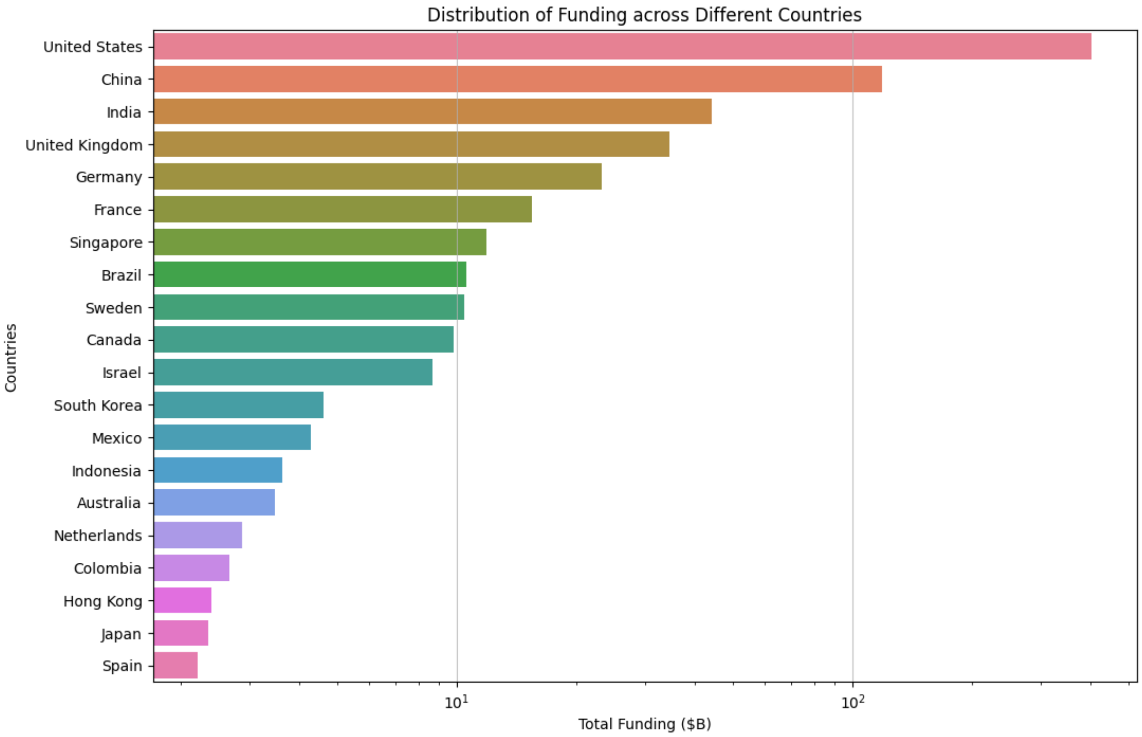
```
plt.figure(figsize=(12, 6))
sns.barplot(y=industry_funding_df['Industry'], x=industry_funding_df['Total Equity Funding
↪ ($)'], hue=industry_funding_df['Industry'], palette='husl')
plt.title('Distribution of Funding across Different Industries')
plt.xlabel('Mean Funding ($)')
plt.ylabel('Industry')
plt.grid(axis='x', alpha=0.75)
```



Distribution of Funding across Different Countries

```
# Group by Country and sum valuations
country_funding_df = df.groupby('Country')['Funding
↪ ($B)'].sum().reset_index().sort_values('Funding ($B)', ascending=False).head(20)
country_funding_df
```

```
plt.figure(figsize=(12, 8))
sns.barplot(y=country_funding_df['Country'], x=country_funding_df['Funding ($B)'],
↪ hue=country_funding_df['Country'], palette='husl')
plt.title('Distribution of Funding across Different Countries')
plt.xlabel('Total Funding ($B)')
plt.ylabel('Countries')
plt.grid(axis='x', alpha=0.75)
plt.xscale('log')
plt.show()
```

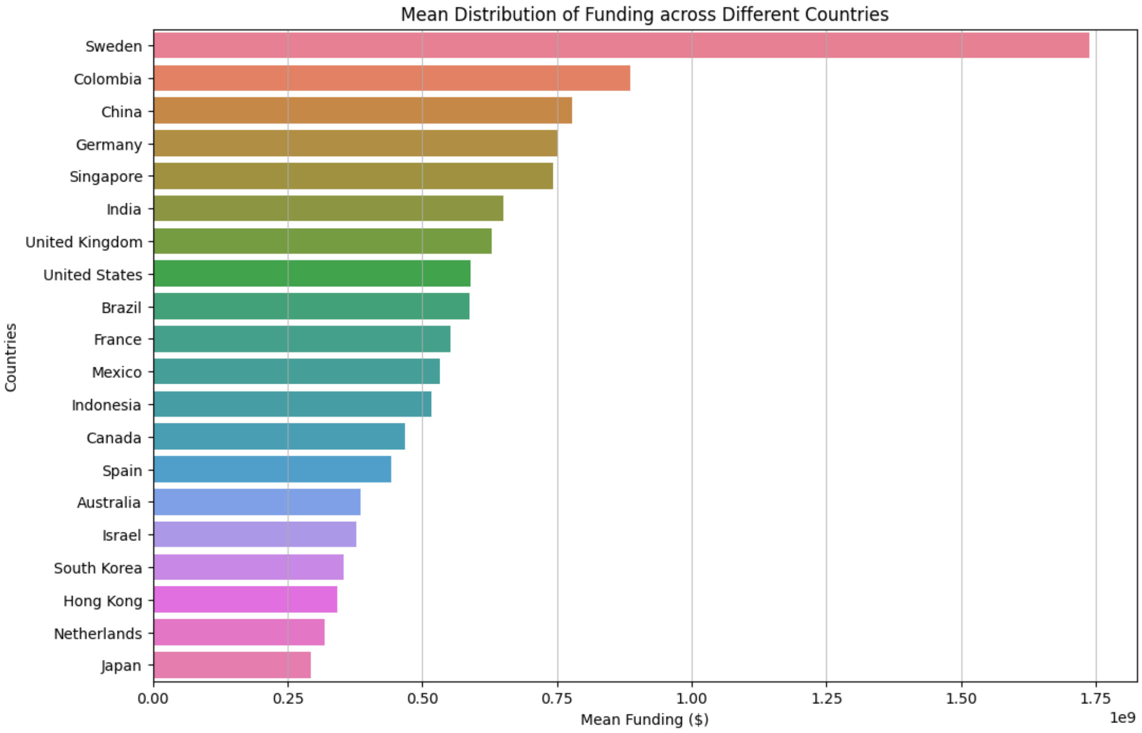


Mean Distribution of Funding across Different Countries

```
# Group by Country and sum valuations
mean_country_funding_df =
↪ df[df['Country'].isin(country_funding_df['Country'])].groupby('Country')['Total Equity
↪ Funding ($)'].mean().reset_index().sort_values('Total Equity Funding ($)',
↪ ascending=False).head(20)
mean_country_funding_df
```

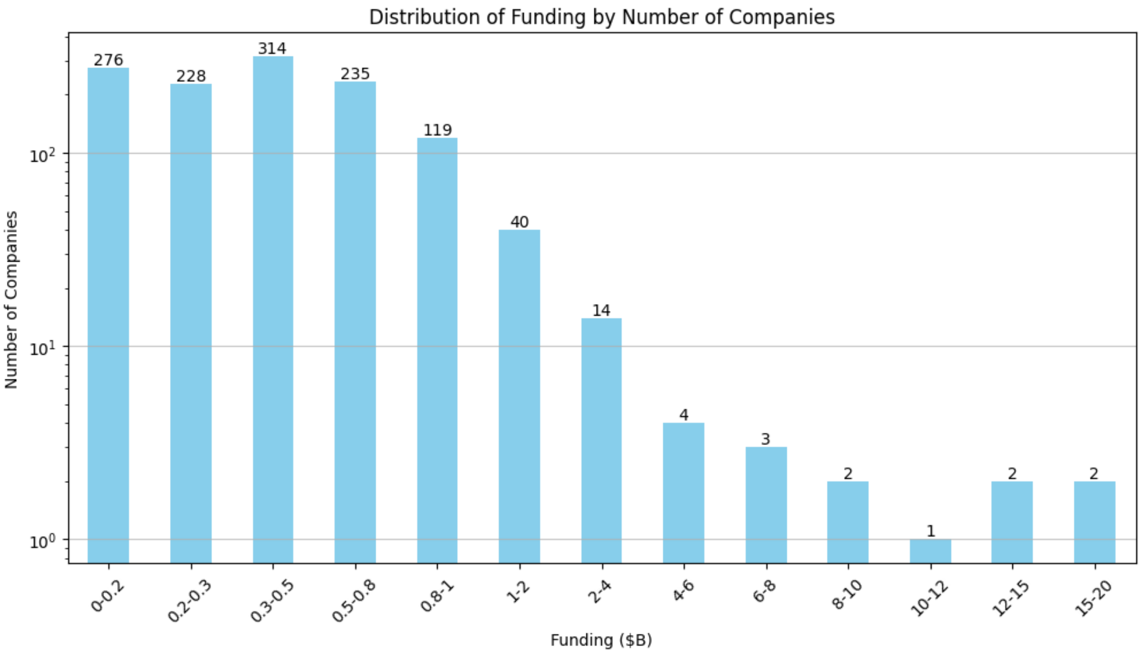
```
plt.figure(figsize=(12, 8))
sns.barplot(y=mean_country_funding_df['Country'], x=mean_country_funding_df['Total Equity
↪ Funding ($)'], hue=mean_country_funding_df['Country'], palette='husl')
```

```
plt.title('Mean Distribution of Funding across Different Countries')
plt.xlabel('Mean Funding ($)')
plt.ylabel('Countries')
plt.grid(axis='x', alpha=0.75)
plt.show()
```



Distribution of Funding by Number of Companies

```
# Define the bins for funding ranges
bins = [0, 0.2, 0.3, 0.5, 0.8, 1, 2, 4, 6, 8, 10, 12, 15, 20]
labels =  [f'{a}-{b}' for a, b in zip(bins[:-1], bins[1:])]
cuts = pd.cut(df['Funding ($B)'], bins=bins, labels=labels)

# Count the number of companies in each bin
funding_distribution = cuts.value_counts().sort_index()

# Plot the Bar Chart
plt.figure(figsize=(12, 6))
ax = funding_distribution.plot(kind='bar', color='skyblue')
ax.bar_label(ax.containers[0])
plt.title('Distribution of Funding by Number of Companies')
plt.xlabel('Funding ($B)')
plt.ylabel('Number of Companies')
plt.xticks(rotation=45)
plt.grid(axis='y', alpha=0.75)
plt.yscale('log')
plt.show()
```



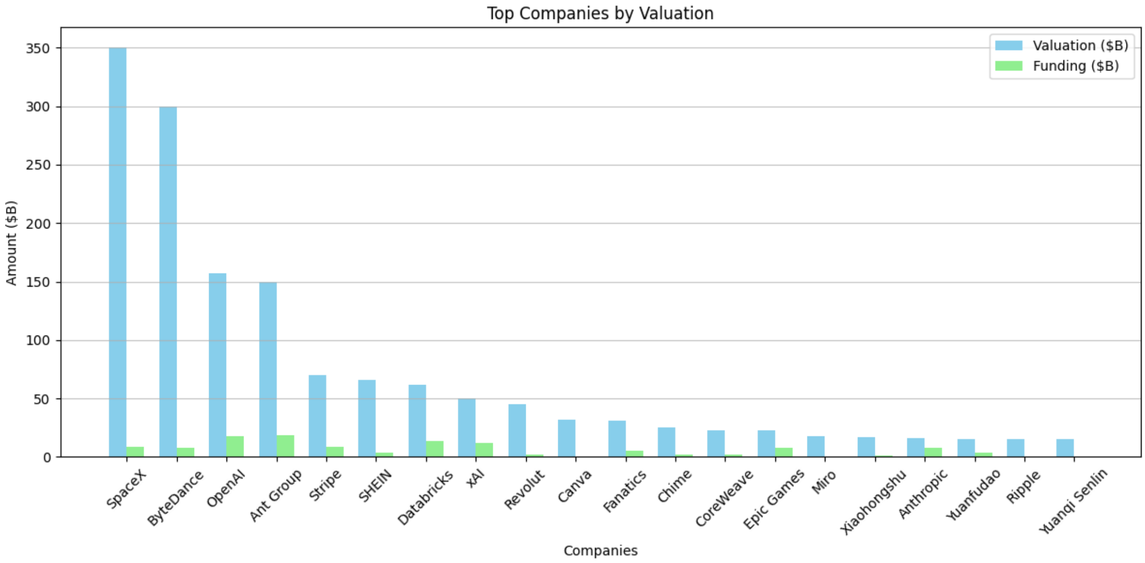
4 Comparative Analysis

4.1 By Company

4.1.1 Top Companies by Valuation

```
top_companies = df.sort_values(by='Valuation ($B)', ascending=False).head(20)
top_companies
```

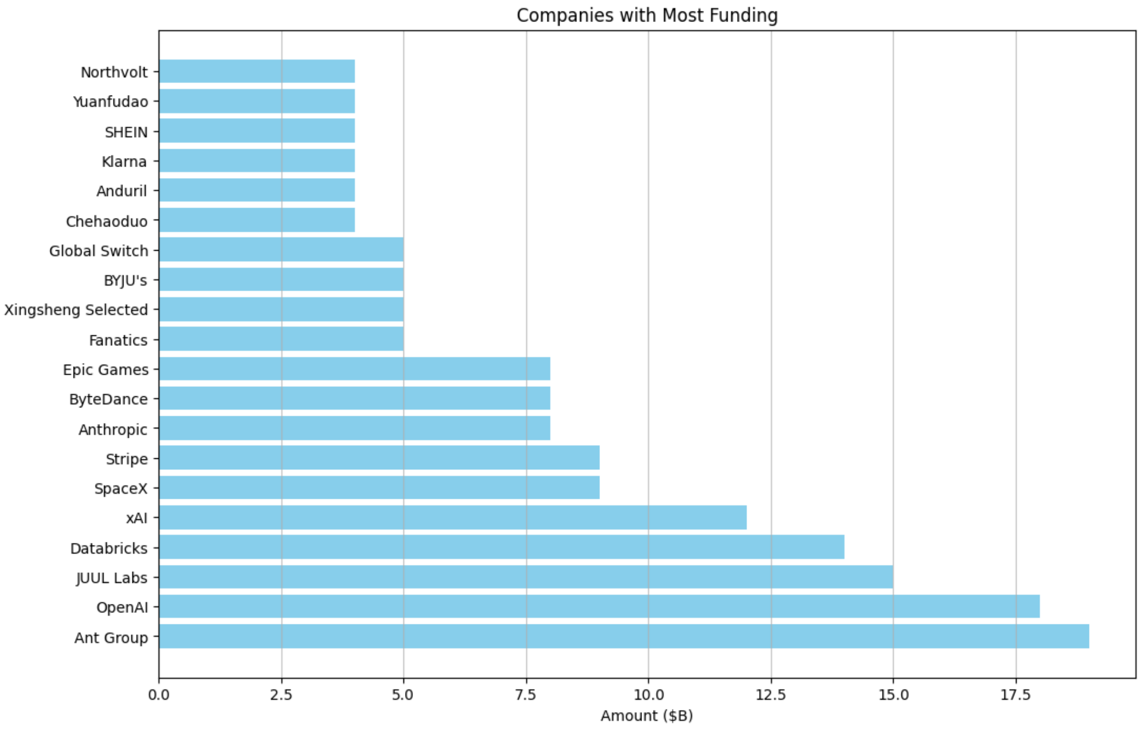
```
# Set the positions and width for the bars
N = len(top_companies)
ind = np.arange(N)  # the x locations for the groups
width = 0.35  # the width of the bars

# Create the bars for valuation and funding
plt.figure(figsize=(12, 6))
bars1 = plt.bar(ind, top_companies['Valuation ($B)'], width, label='Valuation ($B)',
↪ color='skyblue')
bars2 = plt.bar(ind + width, top_companies['Funding ($B)'], width, label='Funding ($B)',
↪ color='lightgreen')

# Add labels and title
plt.title('Top Companies by Valuation')
plt.xlabel('Companies')
plt.ylabel('Amount ($B)')
plt.xticks(ind + width / 2, top_companies['Company'], rotation=45)
plt.legend()

# Add grid
plt.grid(axis='y', alpha=0.75)
plt.tight_layout()
plt.show()
```



4.1.2 Companies Received Most Funding

```
top_companies = df.sort_values(by='Funding ($B)', ascending=False).head(20)
top_companies
```

```
plt.figure(figsize=(12, 8))
plt.barh(top_companies['Company'], top_companies['Funding ($B)'], color='skyblue')
plt.title('Companies Received Most Funding')
plt.xlabel('Amount ($B)')
plt.grid(axis='x', alpha=0.75)
plt.show()
```



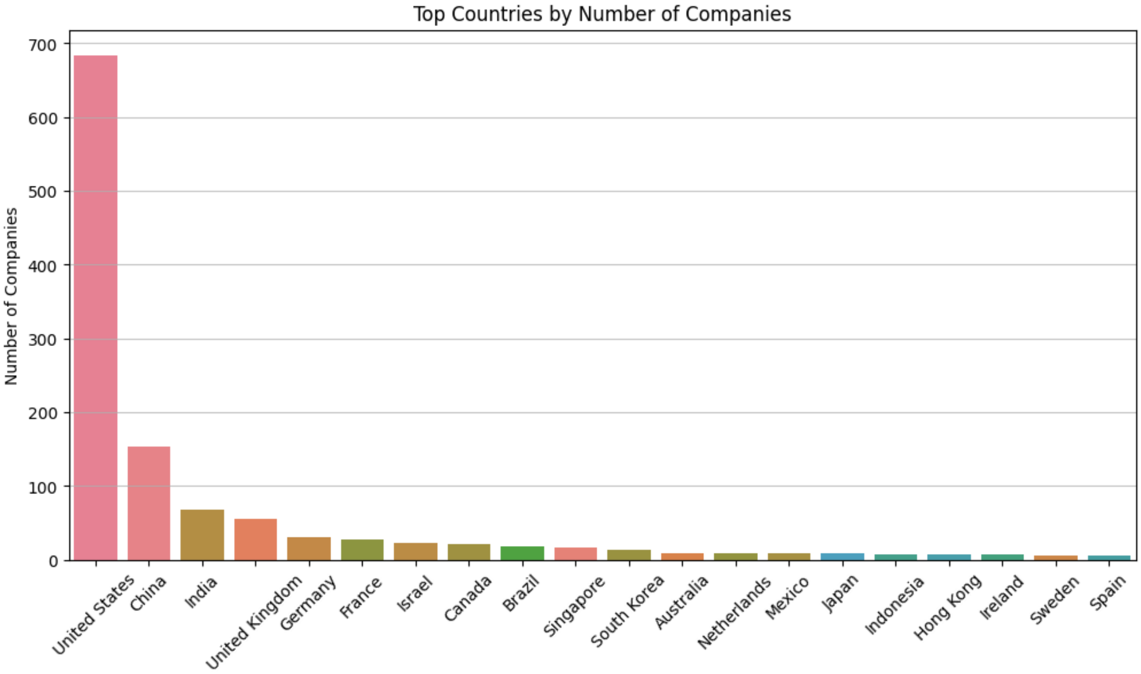
4.2 By Country

```
top_countries = df['Country'].value_counts().nlargest(5).index
top_countries
```

```
Index(['United States', 'China', 'India', 'United Kingdom', 'Germany'], dtype='object',
```

4.2.1 Top Countries by Number of Companies

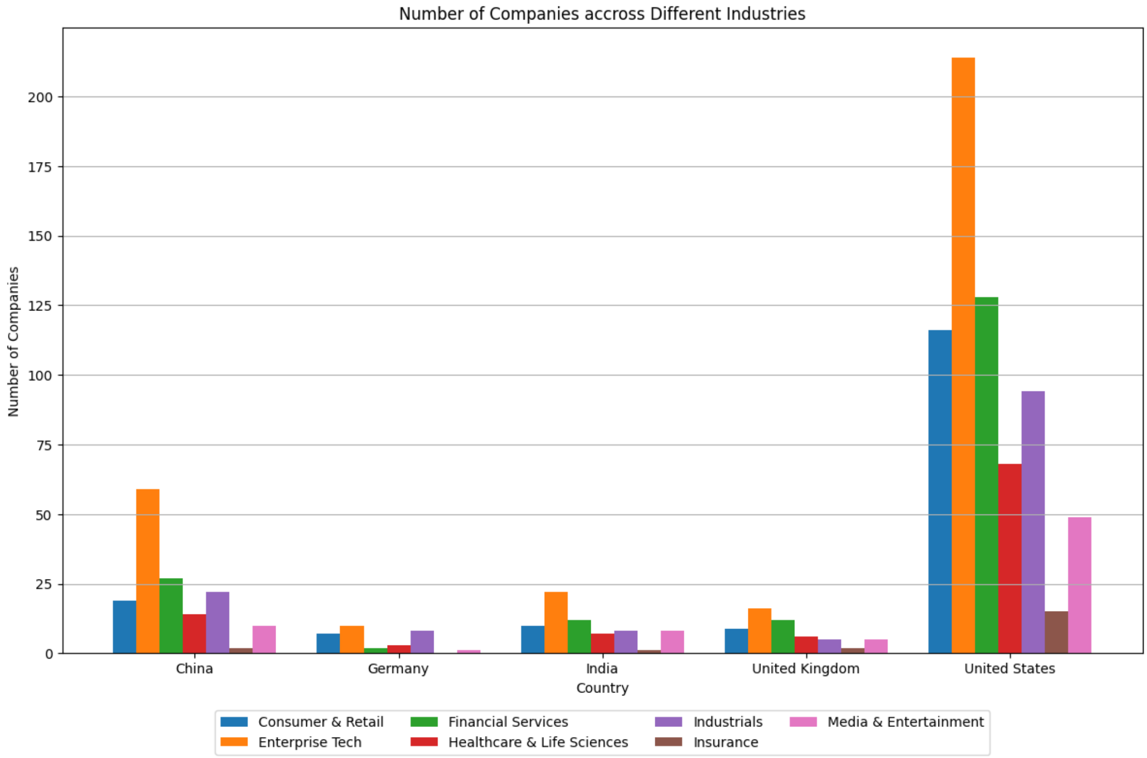
```
plt.figure(figsize=(12, 6))
# sns.barplot(x=top_countries.index, y=top_countries)
sns.countplot(x=df['Country'], order=df['Country'].value_counts().nlargest(20).index,
↪  palette='husl', hue=df['Country'])

plt.title('Top Countries by Number of Companies')
plt.ylabel('Number of Companies')
plt.xlabel(None)
plt.xticks(rotation=45)
plt.grid(axis='y', alpha=0.75)
plt.show()
```



4.2.2 Top Countries by Number of Companies across Different Industries

```
grouped_df = df[df['Country'].isin(top_countries)].groupby(['Country',
↪  'Industry']).size().unstack(fill_value=0)
grouped_df
```

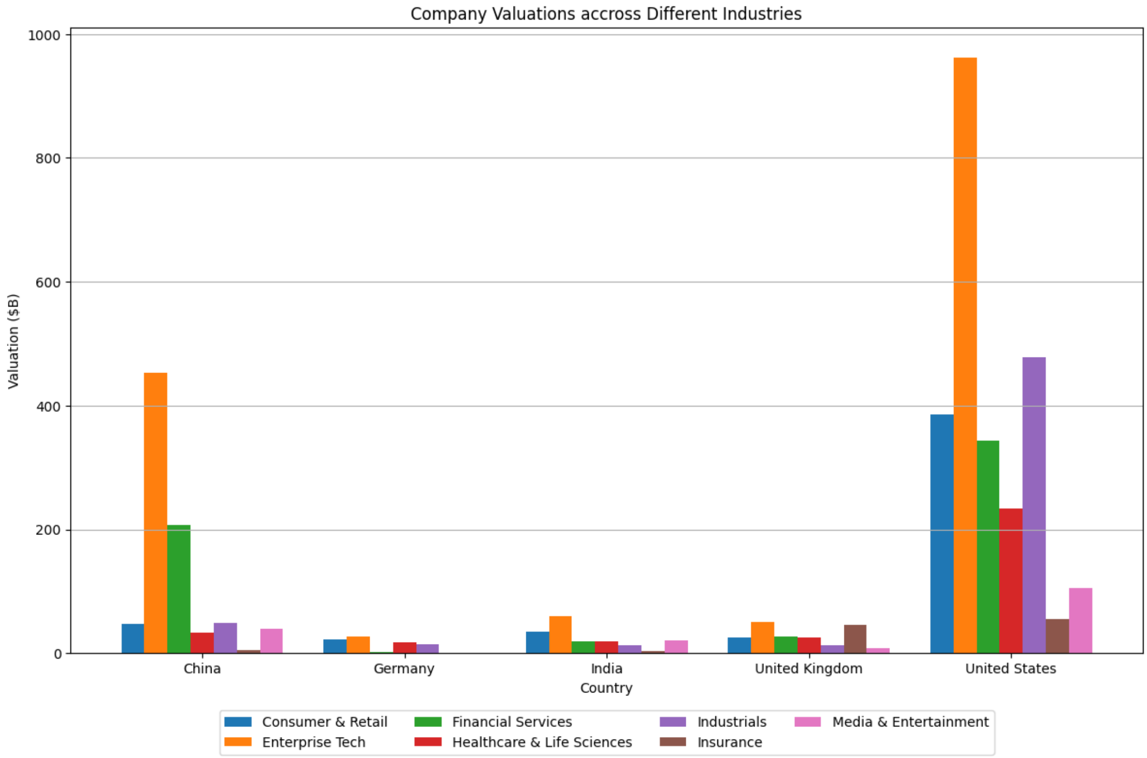
```
grouped_df.plot(kind='bar', figsize=(12, 8), width=0.8)

plt.title('Number of Companies accross Different Industries')
plt.xlabel('Country')
plt.ylabel('Number of Companies')
plt.xticks(rotation=0)  # Keep x-axis labels horizontal
plt.legend(ncol=4, loc="upper center", bbox_to_anchor=(0.5,-0.08))
plt.grid(axis='y')
plt.tight_layout()
plt.show()
```



4.2.3 Top Countries by Company Valuations across Different Industries

```
grouped_df = df[df['Country'].isin(top_countries)].groupby(['Country',
↪  'Industry'])['Valuation ($B)'].sum().unstack(fill_value=0)
grouped_df
```

```
grouped_df.plot(kind='bar', figsize=(12, 8), width=0.8)

plt.title('Company Valuations accross Different Industries')
plt.xlabel('Country')
plt.ylabel('Valuation ($B)')
plt.xticks(rotation=0)  # Keep x-axis labels horizontal
plt.legend(ncol=4, loc="upper center", bbox_to_anchor=(0.5,-0.08))
plt.grid(axis='y')
plt.tight_layout()
plt.show()
```



5 Time-Based Analysis

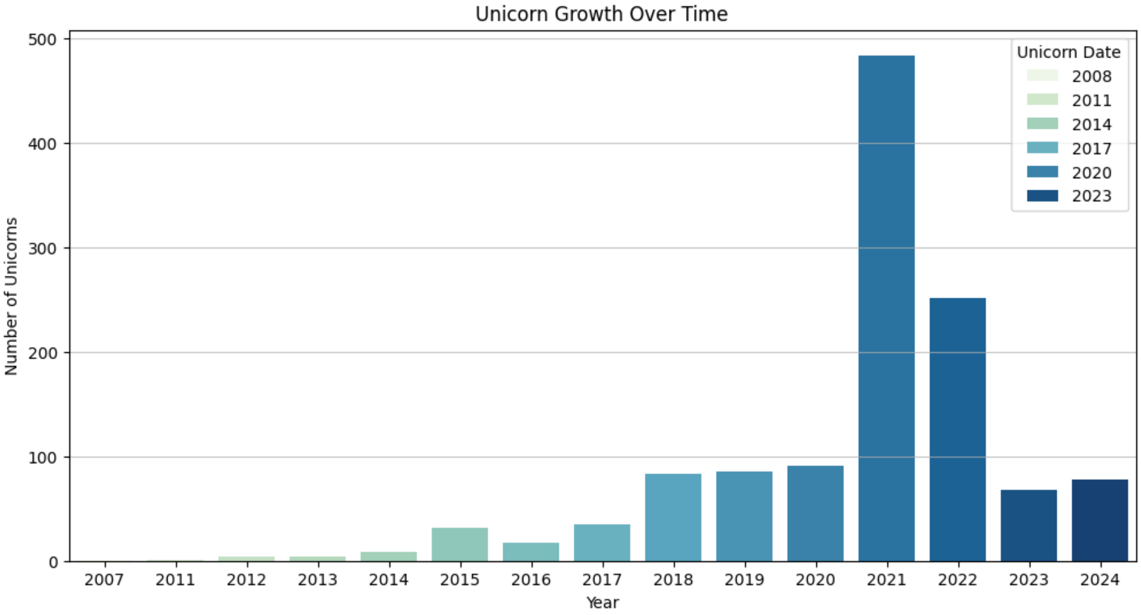
5.1 Unicorn Growth Over Time

```
unicorn_count = df.groupby(df['Unicorn Date'].dt.year).size()
unicorn_count
```

```
Unicorn Date
2007      1
2011      1
2012      4
2013      4
2014      9
2015     32
2016     17
2017     35
2018     83
2019     85
2020     91
2021    484
2022    252
2023     68
2024     78
```

```
dtype: int64
```

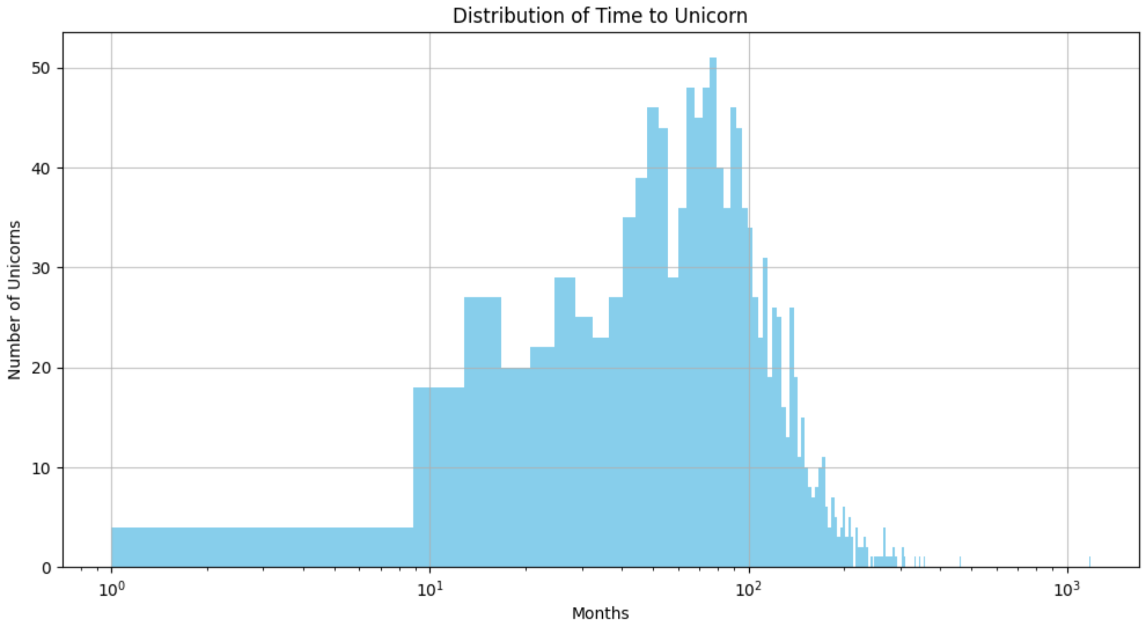
```
plt.figure(figsize=(12, 6))
sns.barplot(x=unicorn_count.index, y=unicorn_count.values, hue=unicorn_count.index,
↪   palette='GnBu')
plt.title('Unicorn Growth Over Time')
plt.xlabel('Year')
plt.ylabel('Number of Unicorns')
plt.grid(axis='y', alpha=0.7)
plt.show()
```



5.2 Time to Unicorn

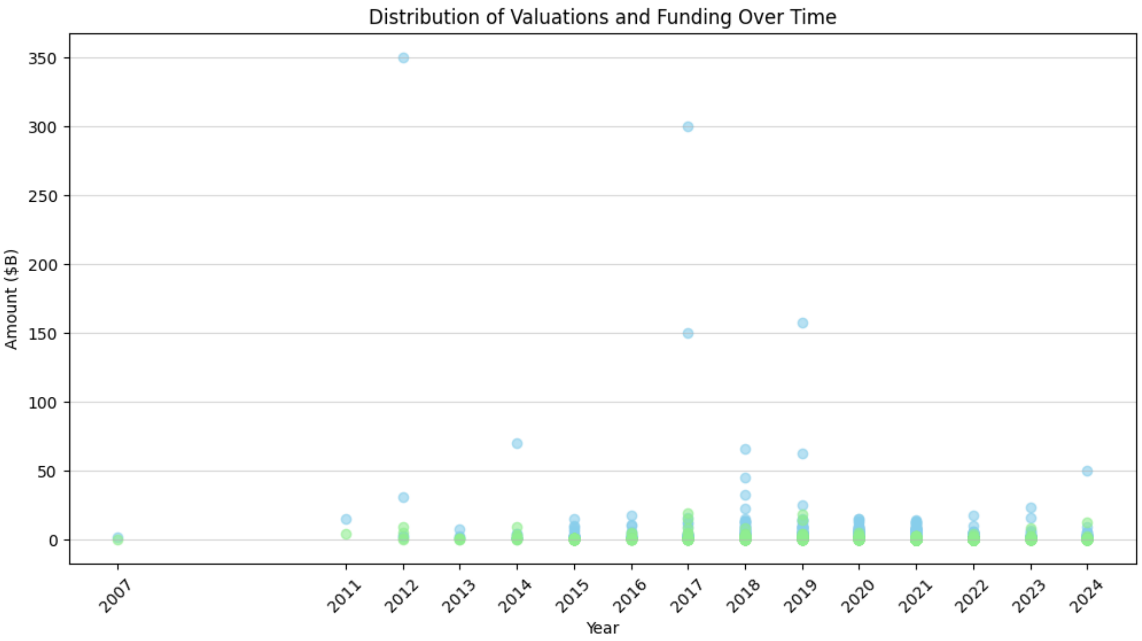
```
# Function to convert "Years to Unicorn" into total months
def convert_years_to_months(years_str):
    if 'y' in years_str and 'm' in years_str:
        years, months = years_str.split('y')
        months = months.replace('m', '').strip()
        return int(years.strip()) * 12 + int(months)
    elif 'y' in years_str:
        years = years_str.replace('y', '').strip()
        return int(years) * 12
    elif 'm' in years_str:
        months = years_str.replace('mo', '').replace('m', '').strip()
        return int(months)
    else:
        return None

df['Years to Unicorn (Months)'] = df['Years to Unicorn'].apply(convert_years_to_months)
```

```
plt.figure(figsize=(12, 6))
plt.hist(df['Years to Unicorn (Months)'].dropna(), bins=300, color='skyblue')
plt.title('Distribution of Time to Unicorn')
plt.xlabel('Months')
plt.xscale('log')
plt.ylabel('Number of Unicorns')
plt.grid(alpha=0.75)
plt.show()
```



5.3 Distribution of Valuations and Funding Over Time

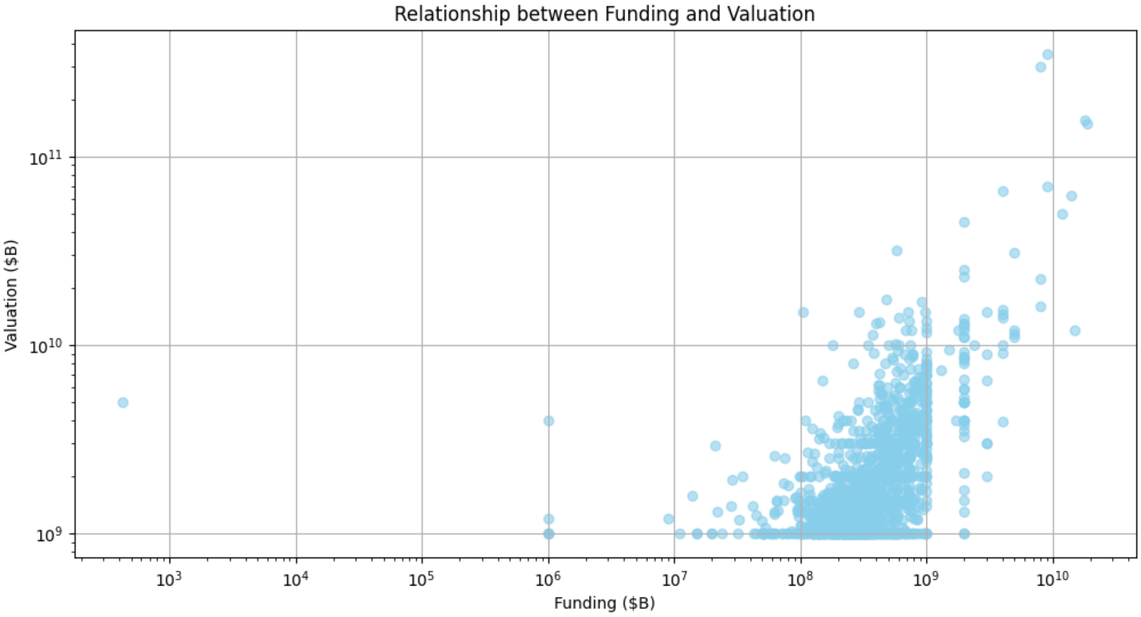
```
plt.figure(figsize=(12, 6))
plt.scatter(df['Unicorn Year'], df['Valuation ($B)'], alpha=0.6, color='skyblue')
plt.scatter(df['Unicorn Year'], df['Funding ($B)'], alpha=0.6, color='lightgreen')
plt.title('Distribution of Valuations and Funding Over Time')
plt.xlabel('Year')
plt.ylabel('Amount ($B)')
plt.xticks(df['Unicorn Year'].unique(), rotation=45)
plt.grid(axis='y', alpha=0.5)
plt.show()
```



6 Correlation Analysis

6.1 Relationship between Funding and Valuation

```
plt.figure(figsize=(12, 6))
plt.scatter(df['Total Equity Funding ($)'], df['Valuation ($B)'] * 1e9, alpha=0.6,
↪  color='skyblue')
plt.title('Relationship between Funding and Valuation')
plt.xlabel('Funding ($B)')
plt.ylabel('Valuation ($B)')
plt.grid()
plt.xscale('log')
plt.yscale('log')
plt.show()
```



7 Historical Analysis

7.1 Survival and Acquisition

1. Find out companies no longer listed in 2024 unicorn list

```
df_2022 = pd.read_csv('input/datasets/Unicorn_Companies (March 2022).csv')
df_out = df_2022[~df_2022['Company'].str.lower().isin(df['Company'].str.lower())]
```

```
179 companies no longer listed in 2024 unicorn list
```

```
df_out.head()
```

2. Financial Stage

```
df_out.size()
```

```
Financial Stage
Acq             1
Acquired        7
Divestiture     1
IPO             2
dtype: int64
```